Desarrollo de un Agente Inteligente Basado en el Estándar ANSI/ISA-95

Conference Paper · November 2014

CITATIONS
0

READS
59

2 authors:

Melina Vidoni
National Scientific and Technical Research Council
21 PUBLICATIONS **12** CITATIONS
SEE PROFILE

Aldo Vecchietti
National Scientific and Technical Research Council
85 PUBLICATIONS **677** CITATIONS
SEE PROFILE

Desarrollo de un Agente Inteligente Basado en el Estándar ANSI/ISA-95

Melina C. Vidoni, Aldo R. Vecchietti
Instituto de Desarrollo y Diseño, Ingar UTN-CONICET
Santa Fe, Argentina
melinavidoni@santafe-conicet.gov.ar, aldovec@santafe-conicet.gov.ar

Abstract

Los cambios en las organizaciones y la búsqueda de la integración, ha generado una necesidad de estandarizar las estructuras de datos empleadas para compartir información, con el objetivo de aumentar la eficiencia del flujo de información. El estándar ANSI/ISA-95 ha cobrado gran relevancia como un medio para la estandarización y automatización de sistemas empresariales, así como también debido a la estructura de información de manufactura que propone y define en la primera y tercera parte del mismo. Este trabajo propone a GrACED, un agente inteligente basado en conocimiento, que procesa lenguaje natural mediante bolsas de palabras, para analizar y clasificar la estructura de las tablas de las base de datos de los ERP, en las categorías propuestas por el estándar ANSI/ISA-95. El objetivo de GrACED y la propuesta de este trabajo es promover un medio, adaptable, portable y con bases estandarizadas, para analizar de forma automatizada la información que puede contener cada tabla en la base de datos, como así también estudiar la adecuación de dicho ERP al estándar, para lograr el objetivo último de facilitar el estudio del sistema empresarial, favoreciendo su integración con otros sistemas.

1. Introducción

Un importante desafío para las organizaciones es el cambio de sus entornos, lo que implica una alta necesidad de flexibilidad, agilidad, eficiencia y calidad en sus procesos. Debido a esto, la Comisión Europea [1] recomendó la mejora de los procesos de integración a través de su estandarización y posterior automatización. La toma de decisiones integradas y la optimización colaborativa dentro de las empresas, pasó a tener un rol crucial en la interrelación de organizaciones. Con este objetivo en mente, se han desarrollado sistemas tipo MES (*Manufacturing Execution Systems*) o CPM (*Collaborative Production Management*) con un único objetivo en mente: anular la brecha entre los procesos, las comunicaciones y los sistemas ERP (*Enterprise Resource Planning*) [2].

Para alcanzar esta integración, es imperativo definir estructuras de información y herramientas sofisticadas que permitan explotar dichas configuraciones, con el objetivo de mejorar la disponibilidad y comunicación de los datos, más específicamente de manufactura, si lo que se desea es integrar cadenas de suministro. Siguiendo esta línea, se han propuesto muchos estándares para mejorar la eficiencia y el flujo de la información de manufactura, entre ellos el ANSI/ISA-95 [3].

ANSI/ISA-95 es un estándar internacional para desarrollar interfaces automatizadas entre empresas y sistemas de control, que propone un conjunto de modelos y definiciones fundadas en una terminología consistente, para describir las tareas e información de manufactura y producción que deben ser intercambiadas en sistemas que se interrelacionan [4]. En los últimos años, este estándar ha sido ampliamente aceptado, debido a que especifica un modelo funcional completo [5].

Se han realizado varios trabajos académicos para favorecer el intercambio de información estandarizada según los modelos del ANSI/ISA-95, así como diferentes formas de implementación. En 2009 [2] varios autores propusieron una plataforma para el intercambio de información utilizando diagramas BPMN (*Business Process Model and Notation*) basados en los modelos del ANSI/ISA-95. También se han efectuado avances en el área de simulación, con el objetivo de generar especificaciones para desarrollar sistemas uniformes [6]. Otros autores [3] han empleado ontologías para generar un *framework* que integra la toma de decisiones, utilizando las estructuras del ANSI/ISA-88. Finalmente, He y otros (2012) [7] realizaron una herramienta para el modelado de empresas, con fundamentos en el ISA-95 y en el IEC 62246.

Sin embargo, si bien muchos estudios se enfocan en diseñar nuevos sistemas y herramientas basados en el ANSI/ISA-95 [8], muy pocos intentan analizar los ya existentes y proveer un informe sobre su adecuación al estándar, o estudiar qué tipo de información de manufactura contienen, a la luz de la clasificación propuesta en el ANSI/ISA-95. La realización de esto favorecería la integración entre sistemas sin obligar a las empresas a cambiar radicalmente su forma de trabajar, como así también proveer un marco para el análisis de los mismos y posibles formas de modificarlos -con el objetivo de adecuarse al ANSI/ISA-95.

A su vez, en un desarrollo previo [9], los autores generaron un prototipo de un sistema tipo APS (*Advanced Planning and Scheduling*) y encontraron la necesidad de presentar al usuario alguna indicación sobre la ubicación de la información de manufactura en la base de datos del ERP, con el objetivo de enlazar los modelos matemáticos del APS, con el ERP empleado.

Utilizando esta idea como disparador inicial, la propuesta detallada en este artículo es utilizar un agente inteligente, cuya base de conocimiento esté dada por el ANSI/ISA-95, y que pueda clasificar el contenido de una base de datos de un ERP o sistema empresarial, en cada una de las categorías que el estándar propone, empleando el enfoque *bag of words* (o bolsas de palabras). Una de las fortalezas de utilizar un agente inteligente, es la capacidad inherente del mismo de procesar el lenguaje natural.

En esta línea, también se han realizado proyectos sobre categorización de textos o estructuras, utilizando agentes inteligentes. En uno de ellos, se ha propuesto una clasificación sobre fuentes de generación de gas, utilizando algoritmos genéticos y redes neuronales para posteriormente compararlos [10]. Una investigación relevante generó un método de clasificar documentos de texto de forma automatizada, usando un conglomerado de múltiples agentes que procesaban lenguaje natural; en este enfoque, cada agente sólo catalogaba en una sola categoría [11]. Finalmente, la idea de bolsas de palabras también ha sido empleada, a través de un modelo bayesiano, para la generación de documentos de texto usando agentes inteligentes [12].

Cabe destacar que hasta el momento no se han encontrado trabajos que empleen agentes inteligentes para analizar sistemas existentes a la luz de los conceptos propuestos por el estándar ANSI/ISA-95.

2. ANSI/ISA-95: Base de Conocimiento

Siguiendo la definición de Russel y Norvig [13], un agente inteligente es una entidad autónoma inserta en un ambiente, que percibe lo que sucede en él a través de percepciones (realizadas mediante sensores), y responde a ellas actuando de forma racional, a través de acciones ejecutadas por actuadores. En este punto es importante mencionar que no todos los agentes inteligentes aprenden y actualizan su conocimiento de forma automática; ellos son una especificación de la definición presentada, y se conocen como *learning agents*.

Hay muchos tipos de agentes, entre ellos los *knowledge based*. Estos especializan la definición presentada, ya que poseen una representación del conocimiento y un proceso de razonamiento que lo ejecuta y puede combinarlo con las percepciones del estado actual, antes de seleccionar acciones. Este tipo tiene muchas variantes, ya que puede denotarse como basado en metas, puede o no interactuar con otros agentes, y puede desarrollarse con o sin aprendizaje.

Estos agentes son muy utilizados en el procesamiento de lenguaje natural, dado que su comprensión radica en inferir los estados ocultos, es decir, la semántica detrás de las palabras.

Respecto a los ERP, la persistencia de la información de estos sistemas se realiza en sus bases de datos, las cuales son en su mayoría de tipo relacional. De esta forma, si se quiere analizar cómo se organiza la información en un ERP, es necesario analizar y clasificar la estructuración de los datos en las tablas de su base de datos.

Para poder categorizar, es imperativo tener categorías definidas y estandarizadas, que tengan una aceptación moderada a amplia, y que formen parte de la base de conocimiento (BC) del agente inteligente.

Esto mismo fue lo que llevó a la utilización y aplicación del ANSI/ISA-95, el cual propone de modelos y definiciones consistentes para generalizar la estructura y nombramiento de las tareas e información de manufactura y producción [4]. Más específicamente, en la Parte 3 [14] propone clasificar la información de manufactura en cuatro categorías que definen la información de productos y de producción, las cuales pueden observarse en la Figura 1[1].

De estas categorías, se decidió trabajar sólo con *Product Definition*, *Production Capability* y *Production Schedule*, dado que su composición es más fácil para detectar con el acercamiento propuesto en el trabajo. Esto es consecuencia de que la información de la categoría *Production Response* es a menudo representada sólo en atributos de las tablas y no en tablas completas, lo que

[1] *Es importante mencionar que las categorías se han trabajado en inglés, con el objetivo de mantener los nombres originales del estándar, mientras que las descripciones y otros conceptos serán presentados en español, para una mayor claridad.*

aumentaría la complejidad de la clasificación.

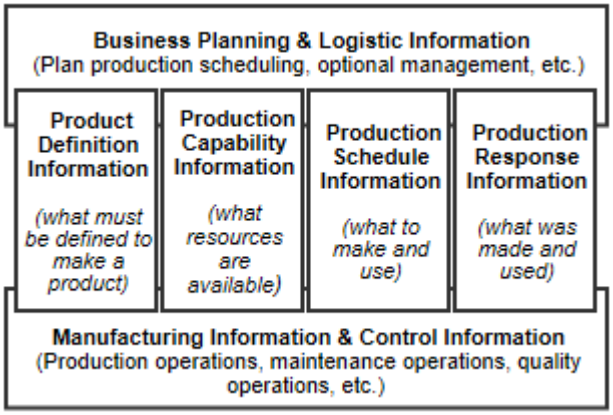

Figura 1.*Categorías de información de ANSI/ISA-95.*

Por otro lado, la Parte I del estándar [4] es la que genera todas las definiciones y conceptos que son posteriormente utilizados para generar la estructura y definir las categorías en las que se va a clasificar. Como parte de estas definiciones, el estándar propone gráficos que serán denominados *de superposición*, que no sólo explicitan subcategorías de información para cada categoría de la Fig. 1, sino que también definen las subcategorías, y cómo se superponen entre ellas.

En la Figura 2 se observa el gráfico de superposición para la categoría *Product Definition*, que es la que posteriormente se ha utilizado en los casos de estudio, con el objetivo de acotar las primeras evaluaciones del agente.

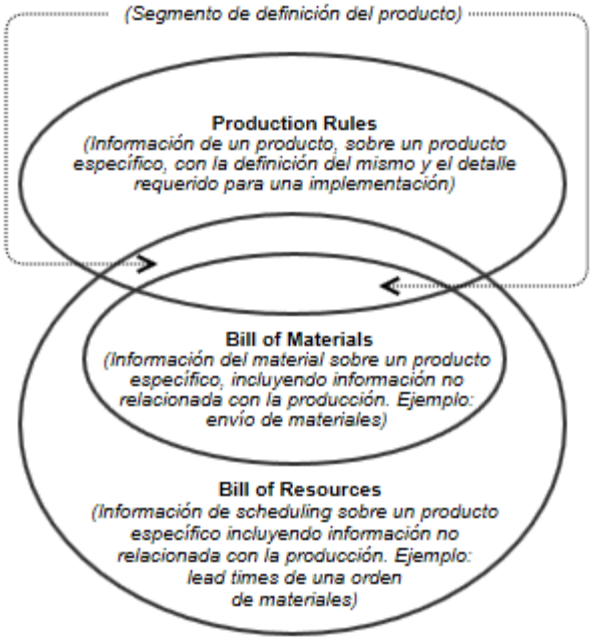

Figura 2.*Superposición de información en la Definición del Producto, para ISA-95 Parte I.*

2.1. Generación de Categorías

De esta forma, para poder generar la BC, el primer paso fue estructurar las categorías y subcategorías de información de manufactura presentadas en el ANSI/ISA-95 en un grafo, el cual puede observarse en la

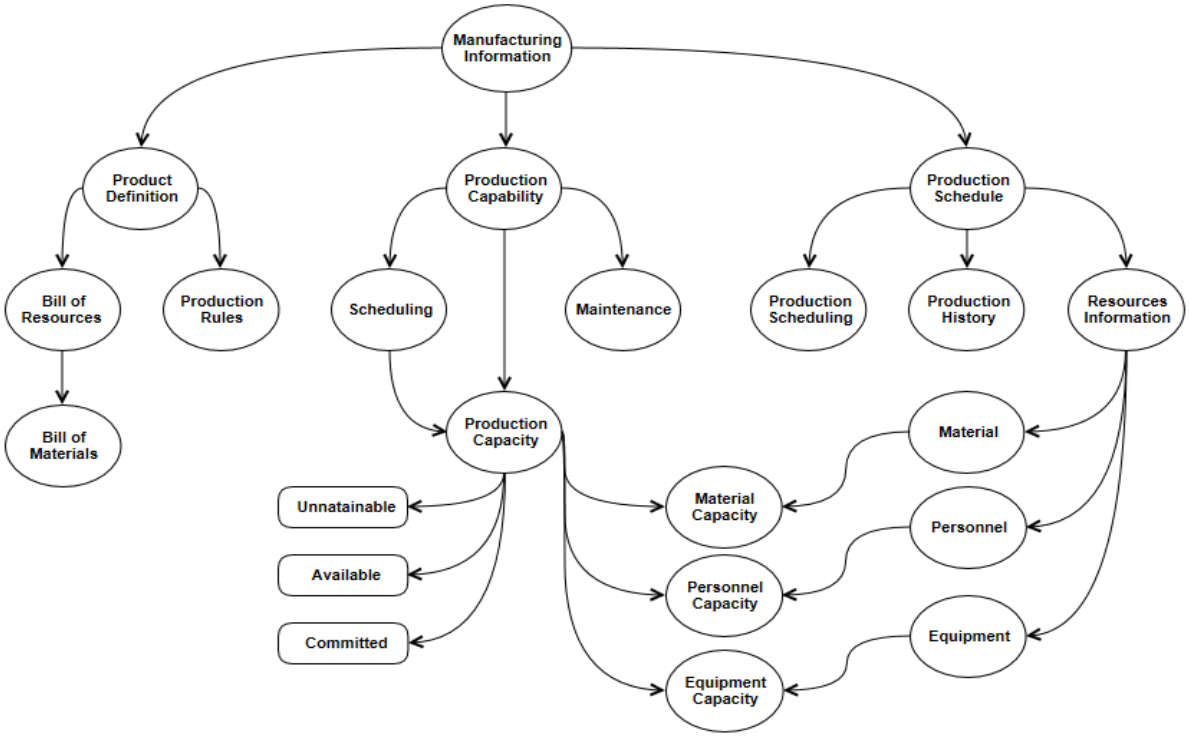

Figura 3.*Grafo de Categorías derivado del ANSI/ISA-95 Parte I.*

Figura 3. Los nodos ovalados representan las categorías (con sus nombres originales en inglés) que pueden utilizarse para clasificar, mientras que los rectangulares de bordes redondeados son presentados en el estándar, pero no se van a emplear en la clasificación.

El nodo raíz representa a la totalidad de información de manufactura, mientras que los nodos de nivel 1 son las grandes categorías de la Fig. 1 (sin usar a *Production Response*, como se mencionó previamente). Por otro lado, los nodos de niveles sucesivos e inferiores fueron obtenidos de los gráficos de superposición (por ejemplo, las categorías visualizadas en la Fig. 2, son representadas como los hijos del nodo *Product Definition* en el grafo de la Fig. 3 y de las descripciones de cada categoría).

2.2. Bolsas de Palabras

El siguiente paso para generar la BC, fue obtener un enfoque que permitiera emparejar las tablas de la base de datos del ERP, con una o más categorías de las presentadas en el grafo. Más allá del Sistema de Gestión de Base de Datos (SGBD) que se emplee, tanto las tablas como las columnas tienen nombres que las identifican, los cuales -generalmente- se eligen para darle significado semántico al contenido que almacenan.

Dado que los ERP son generalmente implementados con BD relacionales, la extracción de las palabras a partir de la definición de sus tablas resulta un enfoque apropiado, dado que dichas estructuras no son propensas a cambiar y fluctuar en el tiempo.

Por esto mismo, se decidió trabajar con lenguaje natural, clasificando las tablas por las palabras que la definen.

El enfoque utilizado es *bag of words (BoW)* o bolsas de palabras. En este enfoque, las bolsas son una representación simplificada utilizada en el procesamiento de lenguaje natural, donde cada clase o documento se representa en un *multi-set* (o bolsa) de palabras, sin considerar la gramática (formación de sentencias) ni el orden de las palabras [12].

En este momento es importante mencionar que el estándar sólo define las categorías, indicando qué tipo de información se incluye, pero sin proporcionar las palabras para formar las BoW. La generación de las BoW fue parte del desarrollo de este proyecto, y será precisado en la siguiente sección.

Sin embargo, no todas las palabras tienen la misma relevancia, la cual incluso puede variar de categoría en categoría; por esto mismo, se decidió asignar pesos a las palabras. Cada bolsa tiene un peso total de 100, el cual fue dividido internamente entre las palabras, dando un peso mayor a las que son más representativas. Además, estos pesos dependen de la categoría que representan.

Un punto importante para mencionar es que, muy a menudo, las palabras que se utilizan para definir nombres de tablas no suelen ser las mismas que se emplean en los nombres de las columnas, aun cuando pertenezcan a la misma categoría. Debido a esto, se decidió asociar dos bolsas de palabras por categoría (o nodo ovalado en el grafo de la Fig. 3): una para las palabras en los nombres de las tablas, y otra para las columnas.

Finalmente, para completar la BC era necesario considerar el uso de sinónimos (palabras escritas de diferente forma pero que significan lo mismo) o abreviaturas (convenciones ortográficas que acortan la escritura de cierto término o expresión) al momento de nombrar las tablas o columnas. Agregar cada combinación para cada palabra a las BoW no es conveniente, ya que no sólo introduce redundancia y aumenta el tiempo de procesamiento, sino que también reduciría los pesos de las palabras dentro de las bolsas; esta redundancia también impactar en el porcentaje de pertenencia final de una tabla a una categoría.

Como consecuencia de la riqueza de los lenguajes naturales, las palabras pueden tener distintos significados, sinónimos y abreviaturas. Estas variaciones dependen de la categoría en la que está siendo clasificada la palabra (por ejemplo, tanto `product` como `production` puede ser abreviado como `prod`). Por esto, se agregaron archivos exclusivos de sinónimos y abreviaturas que fueron relacionados directamente a cada palabra en cada BoW. Estos archivos serán directamente nombrados como *Synonyms Files*, incluso si contienen abreviaturas.

Cabe destacar que, a pesar de la mencionada riqueza de los lenguajes, la cantidad de sinónimos y abreviaturas permanece dentro de un rango manejable para el agente, por varios motivos. Uno de ellos, es que la mayoría de los desarrollos de gran envergadura, tales como los ERP, son codificados en inglés (nombres de variables, métodos, tablas, clases, etc.), el cual es justamente el idioma principal del agente. El otro motivo es que al programar sólo se utilizan sinónimos o abreviaturas convencionales *de facto*, lo que disminuye la cantidad de sinónimos existentes.

2.3. Implementación de la Base de Conocimiento

Como se explicó anteriormente, se asociaron dos BoW por cada nodo del grafo de la Fig. 3. Por esto mismo, se decidió almacenar dicho grafo como el índice de la BC del agente, que contiene las referencias a las categorías y a las bolsas de palabras, pero manteniendo las relaciones de niveles. Esta implementación se realizó utilizando archivos XML [15], y siguiendo la estructura presentada en la Figura 4:

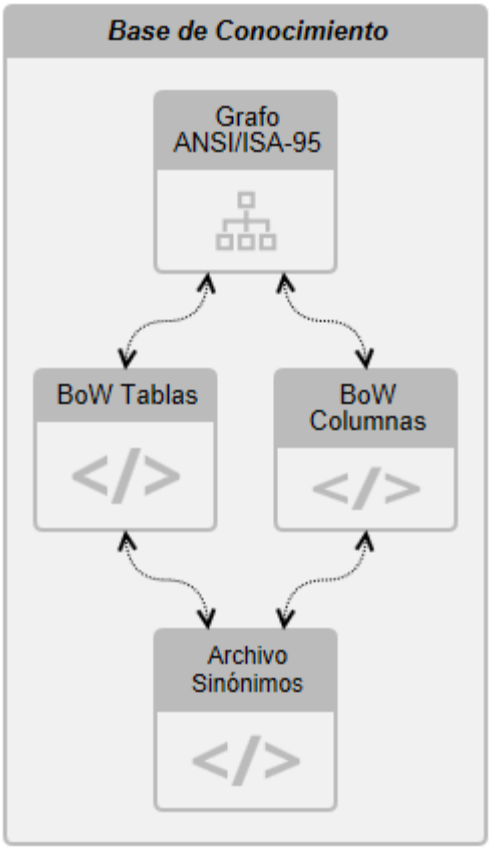

Figura 4.*Estructura de la base de conocimiento propuesta, basada en el ANSI/ISA-95.*

De este modo, debido a que el grafo se guarda como un archivo XML, cada nodo es un elemento dentro del mismo, el cual tiene atributos para el nombre, y para el nombre de archivo de cada bolsa de palabra, diferenciando el uso de cada una. En la Fig. 5 puede observarse un extracto del código XML que representa al nodo *Bill Of Materials*, donde los atributos `columnameBow` y `tablenameBow` son los que guardan el nombre de archivo de las respectivas BoW.

El procedimiento empleado para generar las bolsas de palabras y los archivos de sinónimos fue manual, y se describe a continuación:

1. Se listaron los nombres de tablas y las columnas de cada ERP.
2. Para cada ERP:
    a. Se clasificó manualmente cada tabla, considerando las descripciones de contenido del estándar ANSI/ISA-95.
    b. Se separaron las palabras que conformaban cada nombre de tabla y las de los nombres de columnas. Por ejemplo, el nombre de tabla `stock_inventory_move`, se transformó en tres palabras: `stock`, `inventory` y `move`.
3. Manteniendo la distinción del origen de las palabras (es decir, si eran de los nombres de tablas o de los nombres de columnas) se agruparon todas las palabras de cada categoría (todas las pertenecientes a *Bill of Materials*, las pertenecientes a *Production Rules*, etc.).
4. Para cada grupo de palabras:
    a. Se contó la cantidad de veces que aparecía cada palabra, para obtener la "relevancia" o "nivel de descripción" que aporta la misma para una categoría.
    b. Separadamente, se anotaron cada palabra y los sinónimos de la misma.
    c. Se sumó la cantidad de apariciones de la palabra y sus sinónimos.
    d. Dándole un peso total de 100 a cada BoW, se otorgó un peso a cada palabra, considerando la cantidad de apariciones encontrada en el punto anterior.

3. GrACED: Agente Inteligente

Basándose en la definición de Russel y Norvig presentada al inicio de la sección 2, se propuso un agente inteligente denominado GrACED (por las siglas en inglés de *Grammar Agent for Classifying ERP Databases*). Siguiendo los componentes mencionados endicha definición, en la Figura 6 puede observarse la estructura básica de la propuesta.

De este modo, GrACED representa al agente inteligente propuesto que se encuentra inserto en un ambiente, representado por el ERP que se desea analizar. Este ambiente posee un estado compuesto por los datos de conexión a la base de datos del ERP y la lista de los nombres de las tablas existentes en dicha base, que son las que se van a analizar y clasificar.

Por otro lado, GrACED tiene dos percepciones relacionadas entre sí: obtener el siguiente nombre de tabla a analizar, y luego los nombres de las columnas de dicha tabla. Estas percepciones son almacenadas en el estado del agente, mientras se está ejecutando la única acción que posee: Clasificar. Los otros componentes del estado del agente lo enlazan a la base de conocimiento (la

```
<!--BILL OF MATERIALS NODE-->
<tns:node tns:nodeName="Bill Of Materials" tns:columnameBow="bom_col.xml"
tns:tablenameBow="bom_tab.xml" tns:usable="true">
    <tns:relation tns:relationName="partOf"/>
</tns:node>
```

Figura 5.*Extracto de código XML para el grafo ANSI/ISA-95.*

cual será desarrollado en la siguiente sección) y a una lista temporal para las preclasificaciones obtenidas para la tabla que está analizando mientras la acción Clasificar está siendo ejecutada.

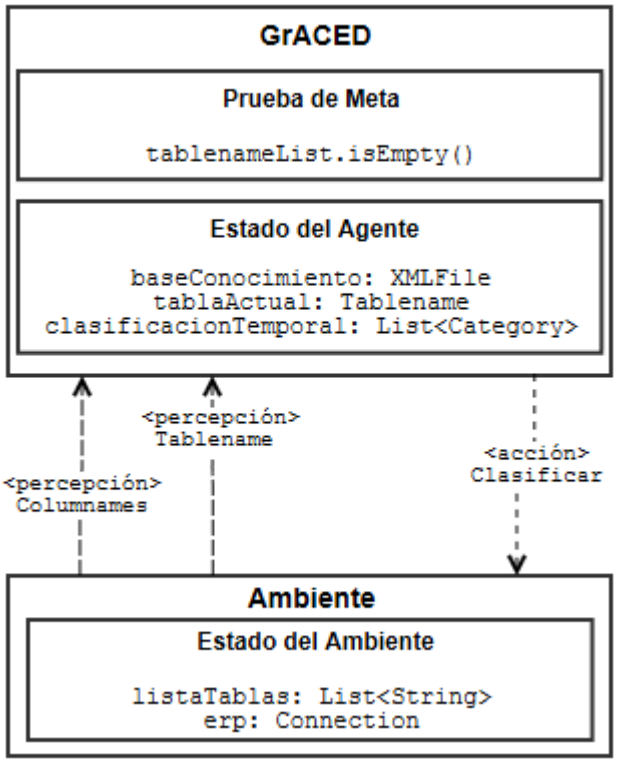

Figura 6. *Estructura básica del agente inteligente.*

Finalmente, el agente también tiene una prueba de meta, la cual le permite evaluar si ha llegado a su objetivo, o si aún necesita continuar trabajando.

3.1. Algoritmo de Razonamiento

El algoritmo de razonamiento es ejecutado durante la acción de clasificar, con el objetivo de emparejar cada tabla con una o más categorías que representen la información que contiene. Para esto, emplea los nodos "habilitados" del grafo, y que sirve para encontrar qué tipo de información almacena cada tabla.

Dado que hay dos BoW por categoría, este algoritmo también tiene dos pasos: el primero, es tomar el nombre de la tabla y clasificarlo utilizando las bolsas de palabras que tiene para ese efecto. Este proceso puede verse en la Figura 7.

Puede notarse que hay dos "filtros" para determinar si una clasificación es adecuada o no. El primero de ellos se realiza comparando la cantidad total de palabras del nombre de una tabla, contra la cantidad de ellas que fueron encontradas en la BoW; para poder pasar a la siguiente etapa, al menos la mitad de las palabras del nombre debe estar en la bolsa. El segundo filtro implica calcular el porcentaje de pertenencia (sumar los pesos de las palabras encontradas), y si dicho valor es menor a un 10%, la clasificación se descarta.

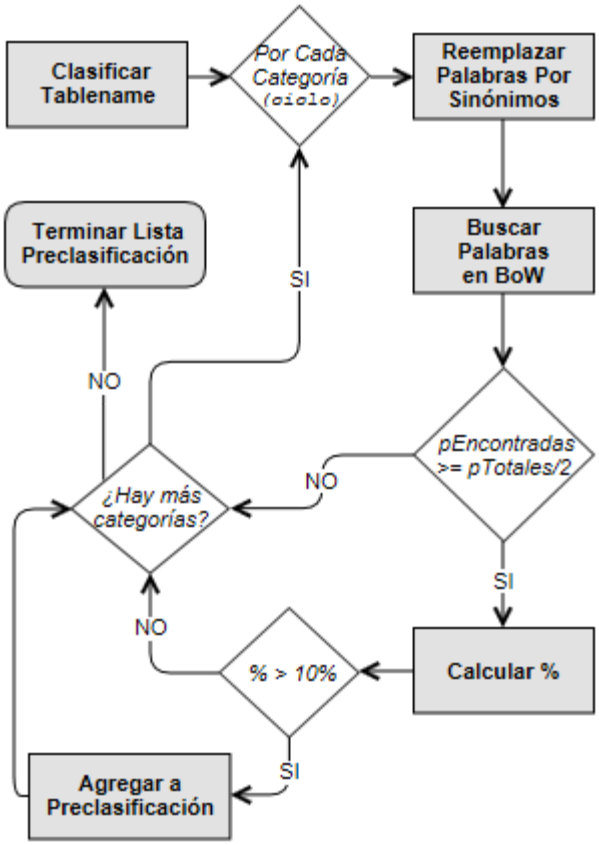

Figura 7.*Preclasificación de tablas – Parte 1 del Algoritmo de Razonamiento.*

El porcentaje del segundo filtro fue seleccionado siguiendo las premisas detalladas a continuación. En promedio, el nombre de una tabla no suele tener más de cuatro palabras, mientras que cada bolsa de palabra para los nombres de tablas debe guardar entre 25 y 30 registros, ya que hay una amplia variedad que puede usarse para cada categoría. Esto hace que encontrar un peso mayor al 10% en el nombre de la tabla, ya genere una clasificación de relativa importancia. Por esto mismo, más adelante se detallarán las tipificaciones que GrACED realiza con las clasificaciones obtenidas.

El segundo paso para la clasificación, es evaluar las palabras en los nombres de las columnas, sólo en las categorías que sobrepasaron la clasificación con el nombre de la tabla. Los pasos seguidos son muy similares a los de la Figura 7, pero con un solo filtro: evaluar que el porcentaje de pertenencia obtenido es mayor que un 15%. Si el filtro no es superado, la categoría es removida de la preclasificación de la tabla. Dado que todas las bolsas pesan 100, y las BoW para las columnas tienen mayor cantidad de palabras (por ende,

menos pesadas), tras evaluar distintas posibilidades, se llegaron a mejores resultados con este valor de filtro.

3.2. Persistencia de los Resultados

La acción del agente inteligente finaliza al persistir los datos de las clasificaciones en dos archivos XML:

- *Tablas Clasificadas:* contiene las tablas que han sido clasificadas en al menos una categoría, el nombre de la categoría, y los porcentajes de pertenencia.
- *Tablas No Clasificadas:* contiene las tablas que no pertenecen a una clasificación. Este segundo archivo existe debido a que no todas las tablas de una base de datos de un ERP contienen información de manufactura.

Por otro lado, GrACED también ofrece gráficos en su interfaz de usuario al finalizar la clasificación, lo que le permite al usuario poder realizar un estudio más complejo de los resultados obtenidos. Dichos gráficos son:

a. Un gráfico de torta con la proporción de tablas que contienen información de manufactura, y las que no. Esto es especialmente útil para analizar la distribución de los datos y la relevancia que cada empresa le da a los mismos.

b. Un gráfico de barras para cada tabla, mostrando las categorías en las que fue clasificada, y el porcentaje que obtuvo al ser analizada por nombre de tabla, por nombre de columna y el promedio. Es decir, que cada tabla puede tener pertenencia a más de una categoría, debido a la ambigüedad del lenguaje natural y a la superposición de los datos.

c. El último gráfico es un diagrama de torta que asigna un tipo a cada uno de los promedios de clasificación, con el objetivo de obtener mayor información respecto a las clasificaciones. Este tema será desarrollado a continuación, con mayor profundidad.

Dado que por cada categoría hay dos bolsas de palabras disyuntas (una para tablas, y otra para columnas), cada una genera un porcentaje de pertenencia distinto. El motivo para esto es que se puede generar un análisis mucho más valioso al evaluar dónde se presenta una pertenencia más fuerte, ya sea en el nombre de la tabla o en el de las columnas. Sin embargo, sí se muestra al usuario el promedio de ambas pertenencias.

Estos dos valores de pertenencia fueron tipificados en 3 tipos. Estos tipos no determinan la cuán correcta es una categorización, sino que estudian la predominancia de palabras genéricas o representativas (o una mezcla de ambas) en la definición de las tablas de la BD. Estos tipos pueden ser:

♦**Falsos Positivos (Tricky):** son clasificaciones en las que la pertenencia del nombre de la tabla es mucho mayor que la obtenida con los nombres de las columnas. Esto sucede en casos donde el nombre de la tabla tiene palabras muy específicas para una categoría, mientras que las columnas tienen palabras genéricas cuyos pesos son medios o bajos. Sin embargo, no se descartan porque más allá de la combinación de pertenencias obtenidas, la tabla puede contener información relevante.

♦**Positivos Totales (True):** representan a aquellas clasificaciones donde ambas pertenencias tienen porcentajes altos, ya que fueron encontradas muchas palabras clave de peso elevado (muy representativas para una categoría). Por lo general, muy pocas tablas por categoría pertenecen a este tipo.

♦**Neutrales (Neutral):** son clasificaciones no englobadas en las anteriores, donde generalmente ambas pertenencias son de nivel medio, y sólo contienen palabras de relevancia intermedia, no muy importantes pero tampoco genéricas. Suelen ser tablas que almacenan información complementaria a las Positivas Totales, tales como tablas que derivan de relaciones del diagrama Entidad-Relación de la BD. La existencia de categorizaciones de este tipo no habla de una mala categorización, sino de que la base de datos emplea muchas palabras genéricas y pocas palabras realmente significativas.

3.3. Separación de Palabras

Como puede notarse, la implementación de un agente que procese lenguaje natural, siempre va a depender de dos situaciones ajenas al mismo: la sintaxis y la semántica de las palabras. Si las palabras son escritas en idiomas que el agente no comprende, o con errores ortográficos, éste no podrá procesarlas. Lo mismo sucede si la semántica de las palabras no es utilizada adecuadamente; por ejemplo, si una columna se llama `cellphone_number` pero en realidad contiene un nombre de persona física, el agente generará una clasificación basada en el nombre de la columna, y no en el contenido de la misma, ya que la semántica de la etiqueta ha sido usada erróneamente.

Un punto importante relacionado con las palabras, es la separación de las mismas. Generalmente, en los lenguajes de programación se utilizan convenciones de nombres (o *naming conventions* por el nombre en inglés), que establecen métodos para separar las palabras. Las bases de datos actuales no tienen ninguna convención preestablecida -y aunque la tuvieran, no hay forma de asegurar que los desarrolladores las utilizarían- por lo que el problema de la separación no resulta trivial.

Para solucionar esto, antes de comenzar la

clasificación, el agente solicita que se le instruya qué método de separación va a emplearse, lo cual puede observarse en la Figura 8.

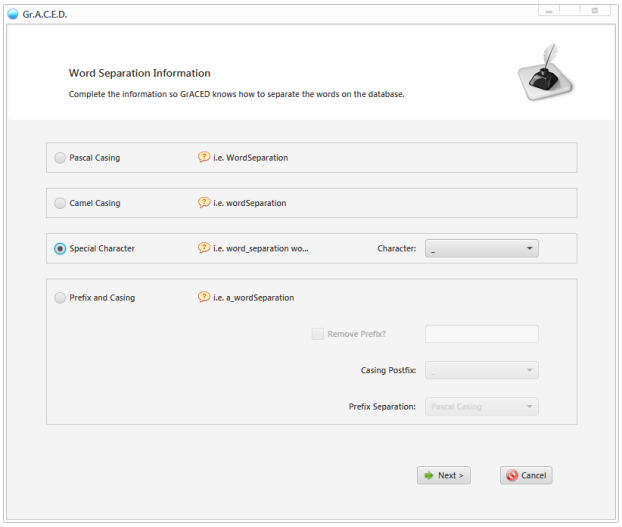

Fig. 8. *Interface gráfica que solicita información sobre la separación de palabras.*

Los tipos de separación comprendidos, por el momento, son:

- ***Pascal Casing*:** las palabras se escriben juntas, y cada una empieza con letra capitalizada. Ejemplo: `UnEjemploDePalabras`.
- ***CamelCasing*:** similar al anterior, la primera palabra lleva letra minúscula. Ejemplo: `unEjemploDePalabras`.
- **Separación por Caracteres:** las palabras son escritas en minúsculas, y cada una se separa usando un carácter especial (punto, espacio, guion medio, guion bajo). Ejemplo: `un_ejemplo_de_palabras`.
- **Separación Mixta:** es una separación más compleja y personalizable, y permite seleccionar un prefijo que será eliminado y no analizado, una separación para el prefijo del resto del nombre, y una para el nombre restante.

Cabe mencionar que si una base de datos no mantiene una semántica adecuada, ni consistencia en el método de separación de palabras, GrACED no clasificará a su máxima capacidad. Esto será demostrado en uno de los casos de estudio analizados en la siguiente sección.

4. Implementación y Casos de Estudio

La implementación de un agente inteligente es una tarea compleja, por lo que se decidió utilizar FAIA [16]: un *framework* generado en Java, que ofrece una estructura de clases abstractas que generan varios tipos de agentes inteligentes (reactivos, basados en metas, basados en conocimiento, etc.), y que sirven de marco para implementar la funcionalidad básica de todo agente (la entidad, el ambiente, estado del ambiente, estado del agente, percepciones y acciones).

A su vez, la base de conocimiento ha sido implementada en XML, como se mencionó previamente, debido a la portabilidad, flexibilidad y universalidad que este lenguaje ofrece, además de permitir una fácil modificación y agregado de palabras. Otra ventaja es que a partir de la versión Java 8, las librerías para la lectura/escritura de este tipo de archivos ya se encuentran incorporadas en el lenguaje, quitando la necesidad de utilizar archivos JAR externos.

Con el objetivo de realizar una primera implementación y evaluar el comportamiento y la arquitectura propuesta para GrACED, se decidió trabajar inicialmente sólo con una rama del grafo de categorías (Fig. 3). De este modo, en la Figura 9 puede observarse el grafo sobre el que se trabajó en la implementación.

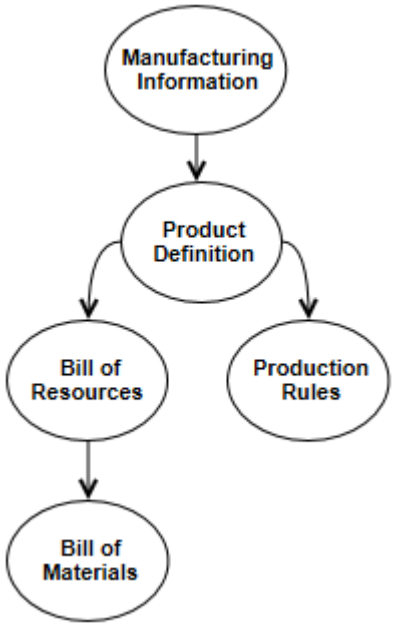

Figura 9.*Implementación inicial del grafo derivado del estándar ANSI/ISA-95.*

Para generar las BoW, los archivos de sinónimos y posteriormente estudiar el comportamiento de GrACED, se emplearon ERP de código abierto pero de amplia utilización en el mercado.

De esta forma, de todos los ERP *open-source* seleccionados para la utilización, sólo se emplearon cuatro para generar la BC del agente (esto incluye BoW de tablas/columnas, y archivos de sinónimos): Compiere [17], OpenERP [18], ERPNext [19] y JFire [20]. Se reservaron dos ERP completamente distintos como conjunto de evaluación: Dolibarr [21] y Libertya [22] (este último no llegó a ser evaluado para este artículo), pero también se agregaron dos casos de estudio

"especiales": OpenERP (ya utilizado en el subconjunto de entrenamiento) y Adempiere [23] (un *fork* de Compiere, pero con bastantes diferencias).

A su vez, el lenguaje natural que se consideró para realizar la BC, fue el idioma inglés. Cualquier palabra en otro idioma, fue tratada como sinónimo de su correspondiente palabra en inglés.

Del grafo de la Figura 8, sólo los nodos *Bill of Materials*, *Bill of Resources* y *Production Rules* fueron empleados para clasificar. A su vez, en la Tabla 1 pueden observarse algunas estadísticas de la base de conocimiento empleada, para poder considerar su tamaño:

Tabla 1. *Datos de la composición de la BC inicial de GrACED.*

Cant. de BoW para Tablenames	3
Cantidad de BoW para Columnames	3
Archivos de Sinónimos	189
Palabras en BoW de Tablenames	70
Palabras en BoW de Columnames	423
Proporción Columnames por Tablenames	6,043
Palabras Totales	493

4.1. OpenERP

OpenERP [18] es una suite ERP de código abierto, publicado con una licencia AGPL2 [24] e implementado como una aplicación web. Su funcionamiento se centra en la lógica de negocios y en el módulo MRP. Esta suite también fue utilizada por los autores en el caso de estudio de una investigación previa [9].

La base de datos de OpenERP fue implementada en PostgreSQL, sí mantiene consistencia en la convención de nombres, usando siempre las letras en minúsculas, separadas con guiones bajos. De esta forma, `m_production_id` fue considerado como un nombre adecuadamente separado (o preciso), mientras que `movementdate` se consideró incorrecto por la falta de guion bajo entre ambas palabras.

Las estadísticas de precisión de la separación de palabras de la BD de OpenERP pueden observarse en la Tabla 2:

Tabla 2. *Datos de Precisión de BD de OpenERP.*

Total de Nombres de Tablas	450
Nombres de Tablas Correctos	416
Total de Nombres de Columnas	4753
Nombres de Columnas Correctos	4652

De este modo, el 92.44% de los nombres de las tablas de la BD se encuentran adecuadamente separados, mientras que el 97.87% de los nombres de columnas tiene una separación correcta. Esto da un total de 97.405% de precisión en la base de datos.

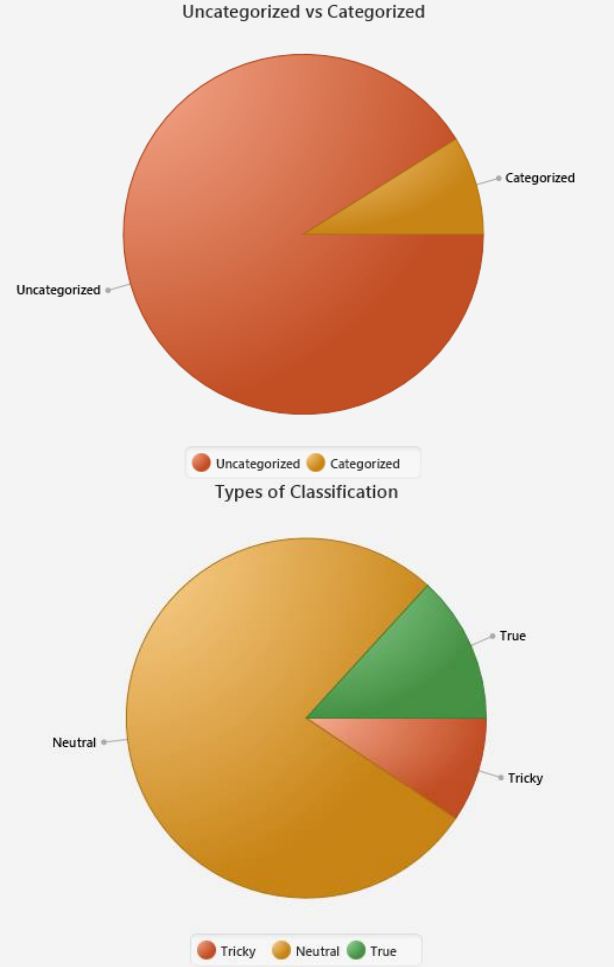

Fig. 10. *Resultado comparando las tablas categorizadas contra las no categorizadas (arriba). Resultado de las tablas categorizadas, separadas por tipo (abajo).OpenERP.*

Este ejemplo fue analizado con GrACED y algunos de los resultados pueden verse a continuación en la Fig. 10 (arriba), donde se puede observar una de las pestañas con resultados generados por el agente, donde del total de tablas, el 8.89% contiene información sobre la categoría *Product Definition* y el 91.11% no. Por otro lado, en la Fig. 10 (abajo) se observa la separación en tipos de clasificación, para todas las tablas que han sido categorizadas. Aquí se cuenta el total de clasificaciones, ya que una tabla puede pertenecer a más de una categoría; de esta forma, hay una mayoría de categorizaciones de tipo neutral (77.36% del total) dado que en la mayoría de las etiquetas o nombres de tablas no se emplean palabras realmente representativas. Del total de categorizaciones, un 13.21% fue tipificada como Positivo Total (pertenencia de nombre de tabla y de nombre de columnas mayor al 50%), y el restante 9.43% se consideró Falso Positivo.

Para evaluar el comportamiento obtenido con GrACED se compararon los resultados automatizados del

agente, contra una clasificación manual realizada por expertos. De esta forma, los expertos realizaron 44 categorizaciones, y GrACED coincidió con 40, lo que representa un 90.91% de certeza. A su vez, el agente agregó 13 categorizaciones, de las cuales 9 fueron posteriormente consideradas correctas por los expertos, tras estudiar el contenido de información y palabras de las mismas.

4.2. Dolibarr

Dolibarr [21] es un ERP de código abierto, publicado bajo una licencia GNU General Public License 3.0 [25], orientado a empresas y compañías de tamaño medio. De origen francés, Dolibarr tiene más de 26 módulos, considerando entre ellos un catálogo de productos y servicios, administración de órdenes de venta y producción, envíos, entre otros.

Para este estudio, se empleó la versión estable 3.5.2 liberada en Abril de 2014, y la base de datos fue implementada en MySQL. Cabe destacar que este ERP no fue utilizado para generar la BC de GrACED.

Similarmente a OpenERP, el método principalmente empleado para la separación de las palabras en dicha BD, es el caracter especial guion bajo. Sin embargo, este ERP no posee la misma precisión que en el caso de estudio anterior, y esto puede verse en los datos de la Tabla 3:

Tabla 3. *Datos de Precisión de BD de Dolibarr.*

Total de Nombres de Tablas	176
Nombres de Tablas Correctos	130
Total de Nombres de Columnas	1967
Nombres de Columnas Correctos	1855

De este modo, Dolibarr posee un 73.86% de precisión en la separación de palabras de los nombres de tablas, y un 94.31% de precisión en los nombres de columnas. Esto da una precisión promedio de 92.63%.

Sin embargo, este ERP tiene un detalle que es importante mencionar: muchas palabras en las etiquetas de columnas y tablas fueron escritas en francés, en lugar de inglés. Para analizar la incidencia del idioma francés en la BD, se separaron las palabras manualmente (corrigiendo aquellas separaciones incorrectas) y se obtuvieron los datos de la Tabla 4:

Tabla 4. *Estadísticas del idioma en Dolibarr.*

Total de Palabras en Tablenames	569
Palabras en Francés en Tablenames	126
Total de Palabras en Columnames	3235
Palabras en Francés en Columnames	307

Calculando los porcentajes, hay un 22.14% de las palabras en los nombres de tablas (o *tablenames*) escritas en francés, y un 9.49% en los nombres de columnas (o *columnames*). Analizando el ERP completo, se obtiene que el 11.38% de las palabras empleadas en la BD fueron escritas en francés, en lugar de inglés.

Para trabajar con estas palabras, el procedimiento fue distinguir las palabras en francés y armar una lista con los significados en inglés de cada una, y posteriormente agregar las palabras francesas al archivo de sinónimos de la palabra en inglés. De este modo, no se modificó la base de conocimiento ni las bolsas de palabras, pero se le dio a GrACED la capacidad de comprender (limitadamente) el francés.

Una vez completados los archivos de sinónimos, Dolibarr fue analizado con GrACED y se obtuvieron los resultados que pueden verse en la Figura 11.

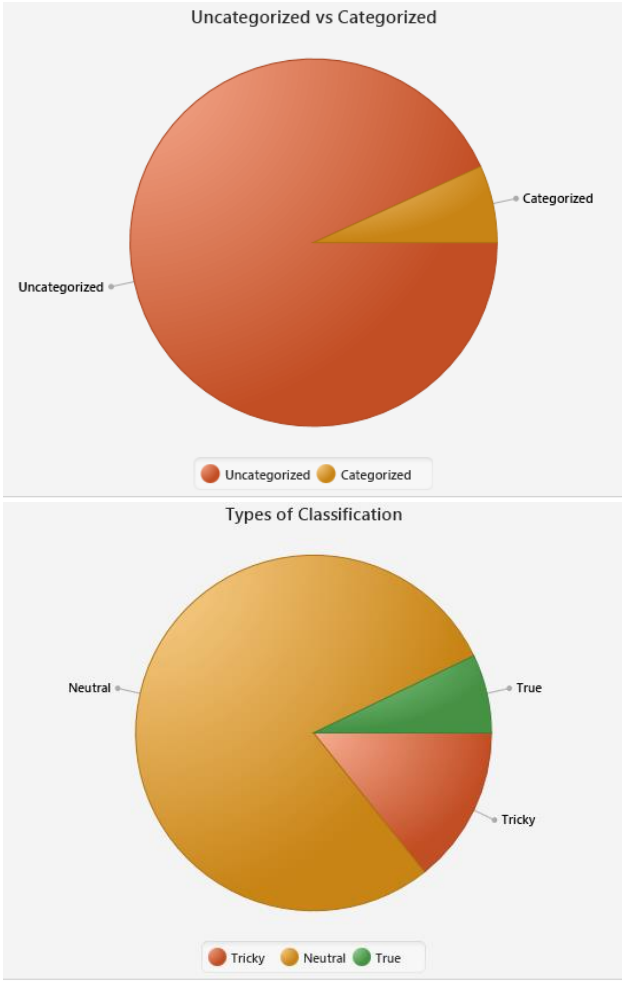

Fig. 11. *Resultado comparando las tablas categorizadas contra las no categorizadas (arriba). Resultado de las tablas categorizadas, separadas por tipo (abajo).Dolibarr.*

De este análisis surge que el 6.82% de las tablas de Dolibarr contienen información de la categoría *Product Definition*¸ que es lo que puede observarse como la porción denominada *Categorized* en la Fig. 11 arriba. Considerando las categorizaciones realizadas en los tipos

definidos previamente, se concluyó que el 78.57% de las son de tipo Neutral, el 7.15% son Positivos Totales (*True*, en el gráfico Fig. 11 abajo) y el 14.28% restante fueron consideradas Falsos Positivos (*Tricky*, en la Fig. 11, abajo).

Nuevamente, y con el objetivo de evaluar el comportamiento de GrACED en el proceso, se compararon los resultados automatizados contra una categorización manual realizada por expertos. Así, los expertos realizaron 16 categorizaciones y GrACED coincidió con 13, lo que representa un 81.25% de certeza. A su vez, el agente agregó sólo 1 categorización, la cual fue posteriormente aceptada como correcta por los expertos.

Este caso de estudio fue considerado como exitoso, si bien es importante destacar la diferencia de tamaño de las bases de datos de OpenERP y Dolibarr, dado que este último tiene una BD de un tamaño 60% menor, aproximadamente. Esto deriva en menor cantidad de tablas con información de Definición del Producto y, por ende, menos clasificaciones.

4.3. Adempiere

Adempiere [23] es otro ERP de código abierto, desarrollo como un *fork*[2] de Compiere, y publicado bajo una licencia GNU General Public License [25]. Este sistema tiene una base de datos de gran tamaño, implementada en Oracle 10g XE.

Este ERP tiene una gran base de datos (con 726 tablas y más de 14000 columnas) pero no posee una buena precisión en la separación de palabras, ya que no ha empleado una convención en particular.

Con los datos de la Tabla 5, se obtiene que sólo 43.52% de los nombres de tablas tienen una separación correcta, mientras que en las columnas el porcentaje es aún menor, apenas alcanzando el 35,36%. El porcentaje total de precisión es35,76%.

Tabla 5. *Datos de Precisión de BD de Adempiere.*

Total de Nombres de Tablas	726
Nombres de Tablas Correctos	316
Total de Nombres de Columnas	14033
Nombres de Columnas Correctos	4958

Esto sucede debido a que no hay una estandarización en la utilización de un *naming conventions*: la única separación que puede considerarse como tal es la utilización del guion bajo, debido a que es la única notable. Oracle es una base de datos sensible a las mayúsculas, y las etiquetas de tablas/columnas han sido escritas totalmente en mayúsculas, aun cuando se podía llegar a emplear *Pascal* o *Camel casing*.

Otro problema en esta base de datos, es el uso de palabras genéricas como "`bname`" o "`description`", sin emplear otros modificadores que agreguen mayor valor semántico, lo que disminuye considerablemente el porcentaje de pertenencia obtenido al intentar clasificar las tablas de esta base de datos.

Finalmente, otro problema de Adempiere es la redundancia de información: se encuentran repetidas gran cantidad de tablas que guardan la misma o similar información, y que sólo agregan datos duplicados, dificultad de mantenimiento y de integración con otros sistemas.

Por lo problemas mencionados es que se decidió analizar Adempiere con GrACED, con el objetivo de poder estudiar el comportamiento del agente en ambientes que no son óptimos. Los resultados de este estudio se encuentran en la Fig. 12.

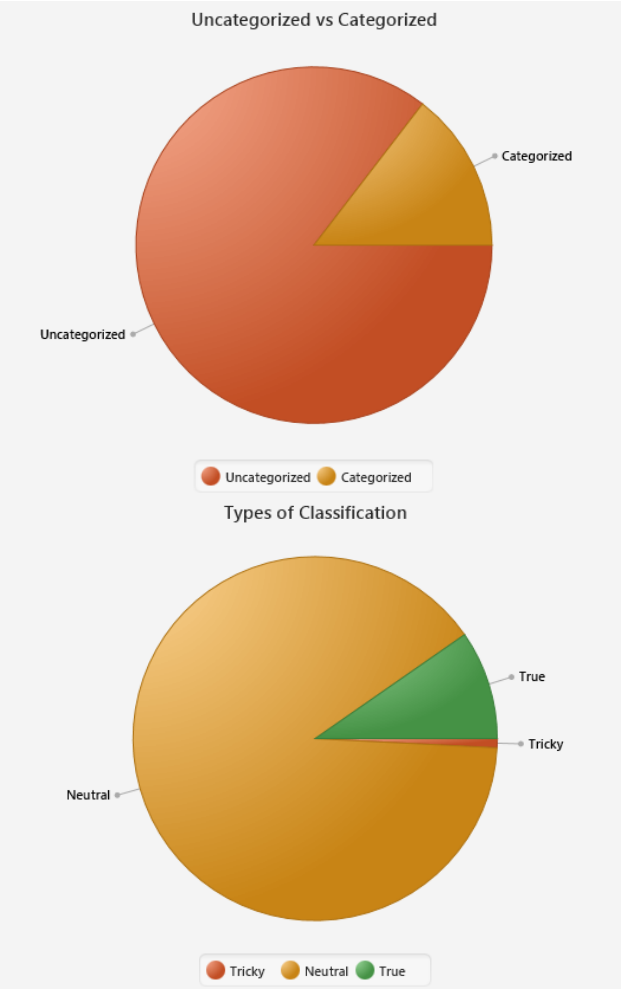

Fig. 12. *Resultado comparando las tablas categorizadas contra las no categorizadas (arriba). Resultado de las tablas categorizadas, separadas por tipo (abajo).Adempiere.*

[2]Un *fork* sucede cuando los desarrolladores copian el código fuente de un paquete de software y comienzan un desarrollo independiente sobre éste, creando un software distinto. Común en desarrollos de código abierto.

De los resultados obtenidos se puede ver que el 14.56% de las tablas de la BD de Adempiere contienen información de *Product Definition*, generando 93 clasificaciones. A su vez, tipifica las clasificaciones de la siguiente forma: el 89.52% son Neutrales, el 98.68% son Positivos Totales y sólo el 0.81% son Falsos Positivos.

Sin embargo, el impacto de la redundancia y la incoherente separación de palabras se ve reflejado en las coincidencias: los expertos sólo clasificaron 30 tablas, de las cuales GrACED coincidió con 19 (un 63.33%). No obstante, el agente agregó 73 clasificaciones y sólo 13 fueron consideradas como adecuadas por los expertos; esto significa que el 65% de las categorizaciones fueron agregados inconsistentes del agente.

Se podría decir que en este caso de estudio, y ante la presencia de problemas de redundancia, carencia de *naming conventions* y utilización excesiva de palabras no representativas, GrACED comienza a "sobre-categorizar" las tablas, en lugar de clasificar de menos. Esto se debe a lo siguiente: un nombre de columna como `ismanufacturingresource` debería contar como 3 palabras pero en realidad, el agente la distingue como una sola palabra porque no la puede separar. Esto hace que la cantidad de palabras totales sea menor que las reales, y se pasen los filtros de cantidad de palabras encontradas.

Podemos concluir que, de cierta manera, es un caso exitoso porque podemos sostener las premisas que se plantearon al inicio: al trabajar en lenguaje natural, existe una fuerte dependencia entre la separación de las palabras y los resultados de la clasificación, así como también entre la semántica de las palabras empleadas y las BoWs generadas.

5. Conclusiones

El presente trabajo propone la estructura básica para un agente inteligente basado en conocimiento y denominado GrACED, el cual trabaja con lenguaje natural (idioma inglés) y que utiliza una base de conocimiento estructurada en bolsas de palabras y generada a partir de la estructura de datos, modelos y definiciones de categorías de información de manufactura propuestos en el estándar ANSI/ISA-95.

El objetivo de GrACED es enlazarse con un sistema ERP (reconocido como su ambiente) y analizar el contenido de su base de datos, para estudiar no solo cómo se estructuran los datos, sino también para encontrar la información necesaria para la integración entre sistemas. Esto resulta especialmente útil al momento de la integración de sistemas de los miembros de una cadena de suministro, o al intentar lograr la colaboración entre el sistema empresarial y un sistema tipo APS (*Advanced Planning and Scheduling*).

Su funcionalidad fue evaluada a través de tres casos de estudio, empleando sistemas ERP de código abierto: OpenERP, Dolibarr y Adempiere, logrando comportamientos favorables, con una precisión total mayor al 80% en los casos exitosos, y un comportamiento esperado en el caso negativo.

Como prototipo del proyecto, la implementación actual de GrACED ha logrado buenos resultados por lo que surgen varios trabajos futuros, entre ellos, lograr la utilización completa del grafo de clasificación, y evaluar un caso más de estudio: Libertya, un ERP de código abierto, origen argentino y base de datos completamente en español.

Otro punto importante, es lograr la propagación de pertenencia a las distintas categorías. Observando el grafo de la Figura 3, la propuesta es que, utilizando las pertenencias obtenidas en la clasificación básica desarrollada en este trabajo, se pueda propagar el porcentaje hacia arriba en el grafo con el objeto de encontrar el impacto que cada tabla tiene en el total de la información de manufactura contenida en la base de datos. A su vez, dado que una tabla puede pertenecer a más de una categoría, esto serviría para dar mayor información sobre a qué categoría de nivel uno pertenece con mayor intensidad.

Al lograr una pertenencia total, también puede estudiarse la adecuación de la base de datos al ANSI/ISA-95, lo cual es el objetivo último de este proyecto, ya que permitiría aplicar a GrACED para estudiar los ERPs que, por ejemplo, desearían aplicarse a una empresa, u obtener indicaciones sobre dónde se encuentra la información necesaria para un intercambio de datos estandarizado.

6. Trabajos citados

[1] EU-Commission, MANUFACTURE - A vision for 2020. Assuring the future of manufacturing in Europe., Office for Official Publications of the European Communities, 2004.

[2] I. Harjunkoski, R. Nyström y A. Horch, «Integration of scheduling and control - Theory or practice?,» *Computers and Chemical Engineering,* vol. 33, pp. 1909-1918, 2009.

[3] E. Muñoz, E. Capón-García, A. Espuña y L. Puigjaner, «Ontological framework for entreprise-wide integrated decision-making at operational level,» *Computers and Chemical Engineering,* vol. 42, pp. 217-234, 2012.

[4] ISA, ANSI/ISA-95.00.01-2000. Enterprise-Control System Integration. Part 1: Models and terminology, ISBN: 1-55617-727-5, 2000.

[5] L. Prades, F. Romero, A. Estruch, A. García-

Dominguez y J. Serrano, «Defining a Methodology to Design and Implement Business Process Models in BPMN according to the Standard ANSI/ISA-95 in a Manufacturing Enterprise,» *The Manufacturing Engineering Society International Conference, MESIC 2013*, vol. 63, pp. 115-122, 2013.

[6] C. Kardos, G. Popovics, B. Kádár y L. Monostori, «Methodology and data-structure for a uniform system's specification in simulation projects,» de *Forty Six CIRP Conference on Manufacturing Systems 2013*, 2013.

[7] D. He, A. Lobov y J. L. Matinez-Lastra, «ISA-95 Tool for Enterprise Modeling,» de *ICONS 2012: The Seventh International Conference on Systems*, 2012.

[8] D. Brandl, «Business to manufacturing (B2M) collaboration between business and manufacturing using ISA-95,» *Revue de l' electricite et de l' electronique*, nº 8, pp. 46-52, 2002.

[9] M. Vidoni y A. Vecchietti, «E2OL: Sistema de Planeamiento y Scheduling Personalizable e Integrable con ERPs,» de *1º Congreso Nacional de Ingeniería Informática y Sistemas de Información*, Córdoba, 2013.

[10] O. P. Quiñonez-Gámez y R. G. Camacho-Velázquez, «Validation of production data by using an AI-based classification methodology; a case in the Gulf of Mexico,» *Journal of Natural Gas Science and Engineering*, vol. 3, pp. 729-734, 2011.

[11] Y. Fu, W. Ke y J. Mostafa, «Automated Text Classification Using a Multi-Agent Framework,» de *Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*, Denver, Colorado, 2005.

[12] H. M. Wallach, «Topic Modeling: Beyond Bag-of-Words,» de *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburg, PA, 2006.

[13] P. Norvig y S. Russel, Artificial Intelligence: A Modern Approach, Prentice Hall, 2010.

[14] ISA, ANSI/ISA-95.00.03-2005. Enterprise-Control System Integration. Part 3: nActivity models of manufacturing operations management, 1-55617-955-3 ed., ISA, 2005.

[15] W3C Recommendation, «Extensible Markup Language (XML) 1.1 (Second Edition),» 2006. [En línea]. Available: http://www.w3.org/TR/xml11/#sec-xml11. [Último acceso: 01 04 2014].

[16] J. Roa, M. Gutierrez, M. Pividori y G. Stegmayer, «How to develop intelligent agents in an easy way with FAIA,» de *Quality and Communicability for Interactive Hypermedia Systems: Concepts and Practices for Design*, IGI global, ed. Francisco V. Cipolla Ficarra, 2010, pp. 120-140.

[17] A. L. Pretorious, Compiere 3, Birmingham: Packt Publishing Ltd., 2010.

[18] OpenERP S.A., «OpenERP,» 2012. [En línea]. Available: https://www.openerp.com/. [Último acceso: 20 April 2014].

[19] Panorama Consulting Solutions, «ERPNext,» 19 Noviembre 2010. [En línea]. Available: http://panorama-consulting.com/erp-vendors/erpnext/. [Último acceso: 2014].

[20] NightLabs Consulting GmbH, «JFire,» 2011. [En línea]. Available: http://www.jfire.net/. [Último acceso: 2014].

[21] L. Destailleur, «Dollibar ERP/CRM,» 2014. [En línea]. Available: http://www.dolibarr.org/. [Último acceso: 2014].

[22] F. Cristina, M. Mauprivez, M. Nerón Cap, J. M. Castro y F. Bonafine, «Libertya ERP,» 2011. [En línea]. Available: http://www.libertya.org/producto/preguntas-frecuentes. [Último acceso: 2014].

[23] B. C. Pamungkas, «ADempiere 3.4 ERP Solutions,» Birmingham, UK, Packt Publishing, 2009.

[24] GNU Affero, «Affero General Public Licence,» 2007. [En línea]. Available: http://www.gnu.org/licenses/agpl-3.0.html. [Último acceso: 20 April 2014].

[25] Free Software Foundation Inc., «GNU General Public Licence,» 29 June 2007. [En línea]. Available: https://gnu.org/licenses/gpl.html. [Último acceso: 2014 April 20].